World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:17, No:10, 2023

Low-Latency and Low-Overhead Path Planning for In-band Network-Wide Telemetry

Penghui Zhang, Hua Zhang, Jun-Bo Wang, Cheng Zeng, Zijian Cao

Abstract—With the development of software-defined networks and programmable data planes, in-band network telemetry (INT) has become an emerging technology in communications because it can get accurate and real-time network information. However, due to the expansion of the network scale, existing telemetry systems, to the best of the authors' knowledge, have difficulty in meeting the common requirements of low overhead, low latency and full coverage for traffic measurement. This paper proposes a network-wide telemetry system with a low-latency low-overhead path planning (INT-LLPP). This paper builds a mathematical model to analyze the telemetry overhead and latency of INT systems. Then, we adopt a greedy-based path planning algorithm to reduce the overhead and latency of the network telemetry with the full network coverage. The simulation results show that network-wide telemetry is achieved and the telemetry overhead can be reduced significantly compared with existing INT systems. INT-LLPP can control the system latency to get real-time network information.

Keywords—Network telemetry, network monitoring, path planning, low latency.

I. Introduction

BIG data center networks require traffic measurement to get fine-grained, real-time network information for various network management applications [1]. In recent years, the development of in-band network telemetry (INT) [2] has gained tremendous progress and can provide better visibility and stability than traditional traffic measurement techniques. However, existing network telemetry technologies still struggle to solve three challenges arising from the growing number of network management applications, and these requirements can be listed as follows.

Telemetry overhead: To complete network measurements, routers need to forward probe packets and incur additional telemetry overhead. As the network scale expands, the additional overhead becomes higher and degrades the performance of the network [3].

Telemetry coverage: The network telemetry system can obtain global information only when the telemetry coverage covers all network devices [4]. Therefore, in order to troubleshoot the network comprehensively, probe packets need to pass through all devices.

System latency: System latency is the time from when a router uploads network information to when the control plane acquires network information. If the system latency is too large will not be able to meet the needs of some latency sensitive network applications [5].

In recent years, existing network telemetry frameworks have been improved for the three key requirements above, such as INT-path [6] and PINT [7]. However, the existing studies still need further improvement in terms of telemetry overhead and telemetry latency. Considering the holistic overhead and system latency, this paper proposes a low-latency and low-overhead path planning algorithm for in-band network-wide telemetry (INT-LLPP). This paper first designs a network telemetry mechanism. It requires only a set of probe flows to obtain different network information. Then, we build a mathematical model to analyze the overall overhead and latency of the network telemetry with full coverage. Next, INT-LLPP adopts a greedy-based algorithm to find the probe paths with minimal overhead to cover the whole network topology. INT-LLPP balances the lengths of the probe flow paths to control the system latency. The simulation results show that INT-LLPP can reduce the telemetry overhead significantly and can control the system latency to get real-time network information.

The rest of the article is organized as follows: Section II presents the related work. Section III discusses the routing mechanism of INT-LLPP and developes a mathematical model for path planning. Section IV designs the probe path planning algorithm. Finally, numerical results are presented in Section V, while Section VI summarizes the paper.

II. Related Work

In-band network telemetry is an emerging representative of network measurement, which has received much attention in academia and industry in recent years. However, existing solutions focus on improving individual targets, making it difficult to balance coverage, telemetry overhead, and latency at the same time.

How to reduce telemetry overhead has been a hot research topic in network telemetry. Existing research has been effective in reducing network and server stress, such as PINT, and INT-label [7], [8]. Specifically, PINT limits the length of each probe packet and splits the network information into more packets to reduce telemetry overhead [7]. INT-label uses interval-based distributed labelling to label internal device information and enable lightweight network telemetry [8].

In terms of obtaining network-wide information, INT packets can be directly flooded into the network topology [9]. However, a large amount of bandwidth will be wasted as some of the devices are repeatedly probed. Therefore, active network

Penghui Zhang, Hua Zhang, Jun-Bo Wang, Cheng Zeng and Zijian Cao are with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 211111, China (e-mail: {phzhang, huazhang, jbwang, czeng, caozijian}@seu.edu.cn).

This work was supported by the National Key Research and Development Program of China under Grant(2020YFB1807803).

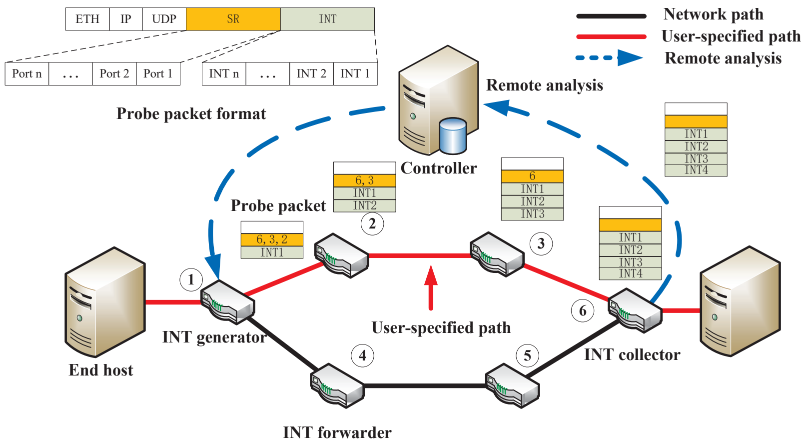

Fig. 1 Source routing-based path monitoring

telemetry (ANT) comes into being. ANT can actively generate probe packets and follow the user-specified forwarding path to collect network information. As long as the forwarding path of probe packets is deployed reasonably, ANT can obtain network-wide information in a stable manner. INT-path [6] and NetVision [10] are representative ANT systems. These extant studies also improve other network performance while ensuring network-wide visibility of telemetry systems. As an example, Euler Trail/Circuit [6] reduces telemetry overhead by generating as few INT packets as possible.

However, these solutions are still not enough for today's network environment. In-band network telemetry as an emerging technology needs more research to achieve better network performance. Since each router supports different network applications, they should insert different kinds of data into probe packets [11]. For example, congestion control requires queue occupancy information and network troubleshooting requires delay information of each node, etc. The existing schemes lack the handling of different telemetry items and lead to unnecessary network overhead. Therefore, considering different telemetry items of each router is a direction of the path planning algorithm design.

III. Design of In-band Network-Wide Telemetry

This section first introduces the routing mechanism of INT-LLPP. Then, it builds a mathematical model for the path planning problem considering the overhead and latency of the network telemetry system.

A. *In-band Network-Wide Telemetry Routing Mechanism*

The routing mechanism of INT-LLPP is shown in Fig. 1. INT-LLPP uses source routing (SR) technology [12] to enable probe packets to be forwarded along a user-specified path [13]. Next, this paper introduces the probe packet format, telemetry operations and network telemetry applications of INT-LLPP in detail.

Probe packet format: The probe packet should allocate one fixed-length stack for the SR label stack in the UDP packet to forward the packet to the specified router port. Specifically, the user-specified paths' router port information is stored in the SR labels, and packets are forwarded by router port ID. In addition, the probe packet should allocate a variable-length INT label stack. Every switch along the path may add an INT label to the packet, which records information on different types of telemetry data. The network telemetry system collects different network data uniformly in the INT label. Therefore, the network telemetry system can complete the data collection for the whole network with only one set of probe packets. For convenience, this paper uses telemetry items to represent the different kinds of telemetry data supported by each route. The size of the INT label depends on the telemetry items supported by the switch. In addition, each INT label should provide the router's latency information to effectively control the system latency in path planning.

Telemetry operations: The telemetry operation of INT-LLPP can be specifically described as a four-step process as follows [6]: (a) The end host needs to generate the empty packets at a fixed frequency and inject it into the data plane. (b) When the INT generator receives the empty packet, it should rewrite the packet header and forward the packet to the next port after adding local network information. (c) The INT forwarder should add local information to the packet and forward the packet according to the user-specified forwarding path. (d) The INT collector is the last router on the forwarding path. After adding local information to the packet, the INT collector needs to forward the packet to the control plane for remote analysis.

Network telemetry applications: In the data center networks, each router may have different telemetry items, such as network throughput and transmission delay. Network telemetry applications require the support of programmable switches. The network telemetry system needs to configure in each switch the kind of network information that needs to be added to the probe packets. Then, each switch processes the probe packets based on the local configuration. It is worth noting that information about the network topology, such as latency, will be collected and used for path planning.

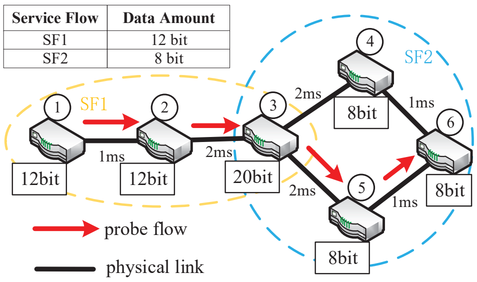

Fig. 2 Service Flow demands to link demands mapping

B. INT-LLPP Path Planning Mathematic Model

We suppose a data center network with n network devices. We define the network topology as an undirected physical graph, denoted by $G = (V, E)$. Each physical node represents a network device, and each connection line represents a network link. $V = \{i | i = 1, \cdots, n\}$ denotes the set of physical nodes, where $i \in V$ represents the index of the physical node. $E = \{e(i,j) | i, j \in V\}$ denotes the set of physical links, where $e(i,j)$ represents the physical link between node i and node j. To facilitate the explanation, this paper uses the case shown in Fig. 2 containing six network devices.

The network telemetry system is assumed to contain M probe flows. The information on the m-th probe flow can be expressed as:

$$f_m = \left[a_1^m, a_2^m, \cdots, a_i^m, \cdots, a_{N_m}^m\right]. \tag{1}$$

where the i-th node of the m-th probe flow is denoted as $a_i^m \in V$. The number of nodes of the m-th probe flow is denoted as $N_m = |f_m|$. $e\left(a_i^m, a_{i+1}^m\right)$ represents the i-th physical link in the m-th probe flow. The set $L_m = \left\{e\left(a_i^m, a_{i+1}^m\right) \middle| i = 1, \cdots, N_m - 1\right\}$, $L_m \subseteq E$ denotes the m-th probe flow's physical links. As shown in Fig. 2, the probe flow passes through nodes 1, 2, 3, 5 and 6 in turn, and can be denoted by $f_m = [1, 2, 3, 5, 6]$. Then, based on the set of links covered by each probe flow, the telemetry coverage of the telemetry system H can be expressed as

$$H = \bigcup_{i=1}^{M} L_i. \tag{2}$$

The set H needs to satisfy $E = H$ for the purpose of obtaining network-wide information.

The system latency is mainly composed of two parts: data transmission and output processing. In order to better serve the network management applications, the system latency cannot be too large. INT-LLPP uses network telemetry from the previous cycle to obtain latency information for each link. The latency function is expressed as $t : E \to T$. Specifically, $t(i,j)$ denotes the latency for a packet from the node i to the node j. If two nodes are not connected, $t(i,j)$ is infinitely large. The latency of the m-th probe flow can be expressed as

$$T_m = \sum_{i=1}^{N_m - 1} t\left(a_i^m, a_{i+1}^m\right). \tag{3}$$

where the latency of the i-th physical link is denoted as $t\left(a_i^m, a_{i+1}^m\right)$.

Since the control plane needs to wait for all probe flows to complete probing before starting remote analysis, the latency of the telemetry system T is determined by the last probe flow to complete its task, $T = \max\{T_m\}, m = 1, 2, \cdots, M$. We assume that the maximum system latency that does not affect the performance of network management applications is $T_{\max}$. In order to ensure real-time network information, the telemetry system must meet

$$T \le T_{\max}. \tag{4}$$

In the process of collecting network information, INT generates telemetry overhead mainly in the control plane and the data plane. The telemetry overhead in the data plane is mainly caused by the forwarding of probe packets, which can be called forwarding overhead c_f^m. Each forwarding by the router incurs forwarding overhead based on the size of the packet. The forwarding overhead c_f^m in the data plane can be expressed as

$$c_f^m = \sum_{j=1}^{N_m} \sum_{i=1}^{j} b\left(a_i^m\right) + N_m b_0, \forall m. \tag{5}$$

where the data size of the packet header is denoted as b_0, the data size of the INT label of the node i is denoted as $b(i)$. In short, the data size that i-th node in the m-th probe flow is denoted as $b\left(a_i^m\right)$, which is determined by the services supported by each router. For example, as shown in Fig. 2, the data size of the third node's INT label can be calculated as $b\left(a_3^m\right) = b(3) = 12 + 8\text{bit} = 20\text{bit}$.

In the control plane, the storage of telemetry data incurs control overhead. Obviously, the control overhead of telemetry packets is related to the size of the packets uploaded to the control plane. The control overhead c_c^m in the control plane can be expressed as

$$c_c^m = \sum_{i=1}^{N_m} b\left(a_i^m\right) + b_0, \forall m. \tag{6}$$

In order to balance the difference between forwarding overhead and control overhead, INT-LLPP takes into account the different system resources allocated in the data plane and control plane and defines forwarding overhead weight h_f and control overhead weight h_c. Therefore, the per-cycle overhead of the m-th probe flow can be expressed as

$$C^m = h_c c_c^m + h_f c_f^m, \forall m. \tag{7}$$

According to (6), the per-cycle overhead of the entire network telemetry system C can be obtained by summing up the entire probe flow's telemetry overhead, which can be expressed as

$$C = \sum_{m=1}^{M} C^m \tag{8}$$

Considering the telemetry overhead and telemetry latency, INT-LLPP establishes a mathematical model for how to design the probe flow path planning under the premise of full network coverage, which can be expressed as

$$\min_{f_m} \; C \tag{9}$$

$$\text{s.t. } T \leq T_{\max} \tag{10}$$

$$E = H. \tag{11}$$

The latency of the network telemetry does not exceed the maximum acceptable system latency is ensured by constraint (10). The detection flow covering the entire network topology is ensured by constraint (11).

The solution of Problem (9) is challenging for the following reasons: (a) the complexity of probing flow path deployment grows as the network size grows. (b) the number and length of paths are unknown, further increasing the complexity of problem solving. In summary, it is extremely complicated to obtain the optimal solution of Problem (9) directly, and an approximation algorithm is used in this paper to solve this problem.

IV. Proposed Low-Latency and Low-Overhead Path Planning Algorithm

In this section, we propose a low-complexity approximation algorithm based on Problem (9) to reduce network telemetry and overhead and latency.

A. Path Planning for INT-LLPP

The purpose of INT-LLPP is low overhead and low latency path planning. The core idea of INT-LLPP is the INT-based greedy algorithm. Specifically, INT-LLPP should select a node after each addition of any node to the probe path. In practice, not all nodes are suitable and a reasonable selection rule needs to be specified. For ease of expression, we assume that at the time of deployment of the m-th probe flow $f_m, m = 1, 2, \cdots, M$, $a^m_{k-1}, k = 2, \cdots, N_m$ has been added into the path f_k, which can be provisionally represented as $f_m = \left[a^m_1, a^m_2, \cdots, a^m_{k-1}\right]$. The nodes connected to the node a^m_{k-1} can be denoted as the set $N\left(a^k_{m-1}\right)$, which is described as

$$N\left(a^m_{k-1}\right) = \left\{i | \exists e(a^m_{k-1}, i)\right\}, \tag{12}$$

where the symbol $\exists$ means physical links exist between nodes. To avoid repeated probing of the same path, the physical link between the node a^m_k and the node a^m_{k-1} should not belong to the set of links have been already covered H, which can be denoted as

$$e\left(a^m_{k-1}, a^m_k\right) \notin H', \tag{13}$$

where $H' = \bigcup_{i=1}^{m} H_i$. Then, the node a^m_k from set $N\left(a^m_{k-1}\right)$ should be selected according to constraint (13). After selecting node a^m_k, the router inserts an INT-label into the packet which is of size $b\left(a^m_k\right)$. Then, the packet should be forwarded $N_k - m$ times until it is uploaded to the control plane. INT-LLPP has developed the selection rule for the node a^m_k to reduce telemetry overhead as follows

$$a^m_k = \arg\min_i b\left(i\right), i \in N\left(a^m_{k-1}\right), \tag{14}$$

According to (5), the INT-based greedy algorithm is adopted. Because the nodes with the smaller INT label are added into the locations that require more forwarding times, selection rule (14) can effectively reduce the packet size during forwarding. To make it easier to understand, take a network with only three routers as an example. For convenience, we assume that the three nodes are numbered 1, 2, 3, and t$b\left(1\right) < b\left(2\right) < b\left(3\right)$. If the nodes are added to the path f_m according to the selection rule (14), f_m can be denoted as $f_k = [1, 2, 3]$, and the forwarding overhead c^m_f can be calculated as $c^m_f = 2b_1 + b_2$. A comparison with the other schemes shows that the telemetry overhead generated by the scheme specified based on the selection rule (14) is the smallest. After adding the node a^m_k into the packet, the path f_m can be updated as $f_m = \left[a^m_1, a^m_2, \cdots, a^m_{k-1}, a^m_k\right]$. Then, we start looking for the next node a^k_{m+1} based on the selection rule (14). It is a special case when choosing the starting node a^m_1 of each path f_m. INT-LLPP can select the most suitable node directly in the set V, which can be denoted as

$$a^m_1 = \arg\min_i b\left(i\right), i \in V, \tag{15}$$

where the node a^m_1 still has links that have not been probed.

Then, no more nodes in set $N\left(a^m_{k-1}\right)$ can be added to the path f_m in the following two cases. First, if all physical links of a node have been covered, then it will not be selected again to avoid duplicate probes. Then, to ensure that the path latency T_m does not exceed the maximum latency $T_{\max}$, INT-LLPP needs to verify $T_m + t\left(a^m_{k-1}, a^m_k\right) > T_{\max}$. Otherwise, the path f_m cannot continue to add node a^m_k due to constraint (4). When in the above two cases, INT-LLPP ends the planning of path f_m and creates a new path f_{m+1}. Finally, to cover all devices in the network, INT-LLPP should verify the telemetry coverage whenever a new node is added, i.e., $E = H'$. After the network-wide information is available, INT-LLPP completes the path planning.

B. Algorithm Description

To facilitate the description of the INT-LLPP path planning algorithm, Algorithm 1 shows the proposed INT-LLPP and the step-by-step procedure is described below:

Step 1) *Initialization:* Create the adjacency matrix to store the network topology and initialize the algorithm parameters. In Algorithm 1, the index of the selected node is represented by the variable v, the current path latency is represented by the variable t, and the data size is represented by the variable b (line 2). According to selection rule (15), create the first probe path and choose the first node (lines 3-5).

Step 2) *Path construction:* Add nodes into the probe paths based on the selection rules (14) (lines 10-12).

In general, the node with the smallest data size should be selected as the starting node of the path.

Step 3) *Creation of new paths:* The variable $flag$ is set to control the creation of new paths (line 9). When the path latency exceeds the maximum system latency or there is no node can satisfy (13), the value of $flag$ should be changed into 0 (line 16). If $flag$ is 0, the planning of the current path should be completed and a new path will be created.

Step 4) *Termination test:* If the network telemetry system covers the entire network topology, or the maximum running time is reached (line 19), the path planning task will be terminated.

Algorithm 1: INT-LLPP Path Planning Algorithm

```
input  : An adjacency matrix A ∈ ℝ^n
output: A set of probe flow paths
1  while L' ⊂ E do
2      v ← 1; t ← 0; b ← b(1);
3      for ∀i ∈ [1, ··· , n] do
4          if b(i) < b then
5              v ← i; b ← b(i);
6      Create a new path and add v into the current path.;
7      while flag = 1 do
8          flag ← 0; b ← 0; v' ← 1;
9          for ∀i ∈ N(v) do
10             if b(i) < b and e(v, i) ∉ L' then
11                 v' ← i; b ← b(i); flag ← 1;
12         t ← t + t(v, v'); v ← v';
13         if t < limit and flag = 1 then
14             Add v into the current path;
15         else
16             flag = 0;
17         Updated the information of L' ;
18         if L' = E then
19             Break;
```

C. Complexity Analysis of INT-LLPP Algorithm

This paper analyzes the run-time complexity of the INT-LLPP algorithm in this section. Only operations related to data processing are considered in this paper, and no other operations are considered, such as allocation. In fact, the network topology G determines how long it takes to find one of the nodes. Algorithm 1 uses the adjacency matrix $\mathbf{A} \in \mathbb{R}^n$ to represent the network topology G. Therefore, the number of nodes directly determines the time to visit each node and the time required to find a connected node. The total time required to complete a path planning is $\mathcal{O}\left(V^2\right)$. Moreover, INT-LLPP should additionally verify $L' = E$ after each node is added to ensure the entire network covered, and requires the additional time $\mathcal{O}\left(V \cdot E\right)$. In summary, the complexity of Algorithm 1 is $\mathcal{O}\left(V^2 + V \cdot E\right)$, and the complexity of the INT-LLPP algorithm does not grow as the topology size increases.

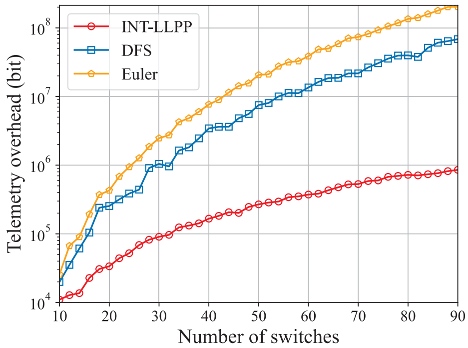

Fig. 3 Telemetry overhead in random topologies

V. Performance Evaluation

A. Simulation Setup

This paper conducts INT-LLPP in Python 3 on the platform with an Intel (R) Core (TM) i7-7700k CPU @ 4.20ghz 4.20GHz machine equipped with 8GB RAM. Then, this paper compares the performance with DFS and Euler Trail/Circuit. Specifically, DFS is a low-complexity full-coverage path planning algorithm. The Euler Trail/Circuit algorithm reduces the telemetry overhead by reducing the number of probe flows. To evaluate the performance of different algorithms, this paper implements INT-LLPP, DFS, and Euler Trail/Circuit [10] in the constructed random network topologies, where the probability that a link exists between nodes is 0.5 in random topology.

Unless otherwise noted, the simulation parameters are set as follows: This paper adopts the latency setting scheme proposed in [14], and each network link latency follows a uniform distribution with a mathematical expectation of 1ms. The data size of the packet header b_0 is 128 bit, and the maximum latency $T_{\max}$ is set as 11ms. Besides, the weight of the overhead on the control plane is set as $w_c = 2$, and the weight of the overhead on the data plane is set as $w_f = 1$.

B. Random Topology Scenario

Random topological scenarios are considered for the experiments, where the links between nodes are randomly generated. The experiments generate five sets of random network topologies with the number of nodes changing from ten to ninety. Then, the results are averaged to reduce the effects of randomness. Next, We summarize the following characteristics:

Telemetry overhead: Fig. 3 shows the telemetry overheads of the three algorithms. It is clear that the telemetry overhead of the three path planning algorithms increases with the size of the random topology. Further, the INT-LLPP algorithm has the least telemetry overhead per cycle. INT-LLPP improves telemetry overhead in several ways. First, large-size labels are forwarded less often due to the selection rules. Then, the restricted path length prevents packets from carrying too much network data and being forwarded multiple times. Finally, INT-LLPP can complete the network telemetry of different telemetry items through a set of probe flows, and this scheme will not increase the number of probe

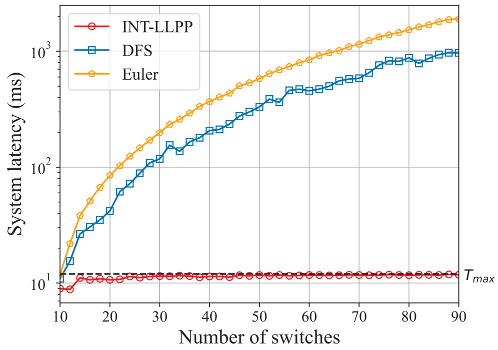

Fig. 4 System latency in random topologies

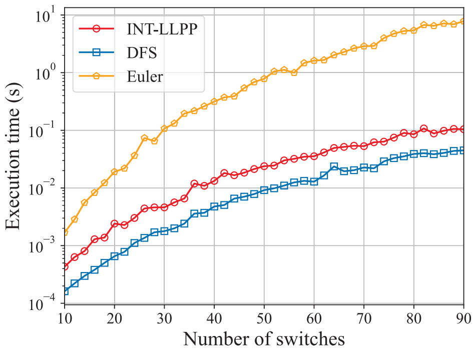

Fig. 5 Execution time of path planning algorithms

flows significantly due to the complex network management requirements. However, INT-LLPP increases the number of probe flows due to the reduced coverage per path, which can result in higher control overhead. Therefore, setting a more reasonable telemetry latency threshold is important to reduce the telemetry overhead, which will be done in further analysis.

System latency in random topologies: Fig. 4 shows the system latency of the three path planning algorithms. It can be seen that the system latency of INT-LLPP is the smallest and INT-LLPP can keep the latency within the maximum latency $T_{\max}$. Compared to the DFS and Euler Trail/Circuit, INT-LLPP is effective in avoiding system latency that grow larger as the number of devices increases. In fact, in a leaf-spine topology or fat-tree topology, DFS and Euler Trail/Circuit will deploy an extremely long probe path to cover most of the network topology, which will increase the system latency. Therefore, INT-LLPP is more suitable for deployment in large-scale networks.

Execution time of path planning algorithms: Fig. 5 shows the execution times for the three algorithms to complete the network telemetry path planning. As shown in Fig. 5, INT-LLPP requires more execution time than DFS, but less than Euler Trail/Circuit. This result validates the runtime complexity analysis.

Impact of the path length on telemetry overhead: Fig. 6 shows the impact of maximum system latency $T_{\max}$ on the telemetry overhead. To reduce the holistic latency of the telemetry system, an effective solution is to reduce the forwarding times on each path by controlling the maximum

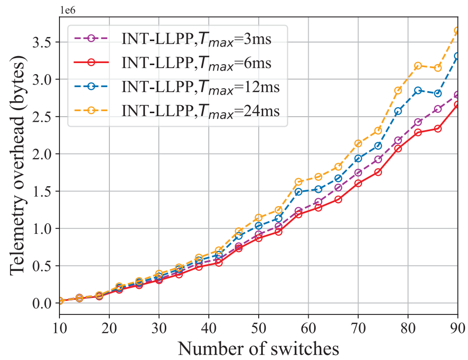

Fig. 6 Impact of maximum system latency on the overhead in different topology scales

system latency $T_{\max}$. However, more paths are required to achieve full network coverage because the area covered by each path becomes smaller. This requires not only more INT agents but also more storage space to store the packets. Thus, there should be a lower limit for the maximum system latency $T_{\max}$, which is constrained by the controller's ability. As shown in Fig. 6, the maximum latency $T_{\max}$ directly affects the telemetry overhead regardless of network scales. When $T_{\max} = 6$ ms, the overhead incurred by network telemetry is minimal. The results show that the most suitable maximum system latency $T_{\max}$ should be chosen according to the network capability to reduce the telemetry overhead.

VI. Conclusions

This paper presents an in-band network telemetry system INT-LLPP that allows for low-latency and low-overhead path planning. This paper builds a mathematical model for probe flow path planning. Specifically, INT-LLPP considers differences in network applications supported by each router to reduce network telemetry overhead. Then, INT-LLPP reduces the system latency by controlling path length. Simulation results show that INT-LLPP can significantly reduce overhead and latency in the data center network topologies. This work requires further consideration of more flexible system latency optimization schemes to better adapt to different networks. Also, the impact of frequency of network telemetry on system performance and the issue of dynamic environment will be left as future work.

References

[1] T. Yang, J. Jiang, P. Liu, Q. Huang, and S. Uhlig, "Elastic sketch: adaptive and fast network-wide measurements," in *the 2018 Conference of the ACM Special Interest Group*, 2018.

[2] M. Yu, "Network telemetry: Towards a top-down approach," *ACM SIGCOMM Computer Communication Review*, vol. 49, no. 1, pp. 11–17, 2019.

[3] T. Barbette, C. Soldani, and L. Mathy, "Fast Userspace Packet Processing," in *2015 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*, 2015.

[4] Y. Lin, Y. Zhou, Z. Liu, K. Liu, Y. Wang, M. Xu, J. Bi, Y. Liu, and J. Wu, "NetView: Towards on-demand network-wide telemetry in the data center," *Computer Networks*, vol. 180, p. 107386, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128620302449

[5] Y. Li, M. Alizadeh, M. Yu, R. Miao, and F. Kelly, "HPCC: high precision congestion control," in *the ACM Special Interest Group*, 2019.

[6] T. Pan, E. Song, Z. Bian, X. Lin, X. Peng, J. Zhang, T. Huang, B. Liu, and Y. Liu, "INT-path: Towards Optimal Path Planning for In-band Network-Wide Telemetry," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, 2019, pp. 487–495.
[7] R. B. Basat, S. Ramanathan, Y. Li, G. Antichi, and M. Mitzenmacher, "PINT: Probabilistic In-band Network Telemetry," *ACM*, 2020.
[8] E. Song, T. Pan, C. Jia, W. Cao, J. Zhang, T. Huang, and Y. Liu, "INT-label: Lightweight In-Band Network-Wide Telemetry via Interval-Based Distributed Labelling," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*. IEEE Press, 2021, p. 1–10. [Online]. Available: https://doi.org/10.1109/INFOCOM42981.2021.9488799
[9] N. Katta, M. Hira, C. Kim, A. Sivaraman, and J. Rexford, "Hula: Scalable load balancing using programmable data planes," in *ACM*, 2016.
[10] Z. Liu, J. Bi, Y. Zhou, Y. Wang, and Y. Lin, "Netvision: Towards network telemetry as a service," in *2018 IEEE 26th International Conference on Network Protocols (ICNP)*, 2018, pp. 247–248.
[11] D. Bhamare, A. Kassler, J. Vestin, M. A. Khoshkholghi, and J. Taheri, "Intopt: In-band network telemetry optimization for nfv service chain monitoring," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–7.
[12] Sunshine and A. Carl, "Source routing in computer networks," *ACM SIGCOMM Computer Communication Review*, vol. 7, no. 1, pp. 29–33, 1977.
[13] Cole, Schlesinger, David, Walker, Amin, Vahdat, Dan, Daly, George, and Varghesex, "P4: Programming protocol-independent packet processors," *Computer Communication Review: A Quarterly Publication of the Special Interest Group on Data Communication*, vol. 44, no. 3, pp. 87–95, 2014.
[14] A. El-mekkawi, X. Hesselbach, and J. R. Piney, "Evaluating the impact of delay constraints in network services for intelligent network slicing based on skm model," *Journal of Communications and Networks*, vol. 23, no. 4, pp. 281–298, 2021.